AUTOSAR Methodology

Contents

1. Overview
2. AUTOSAR Exchange Formats
3. Development of the functional software

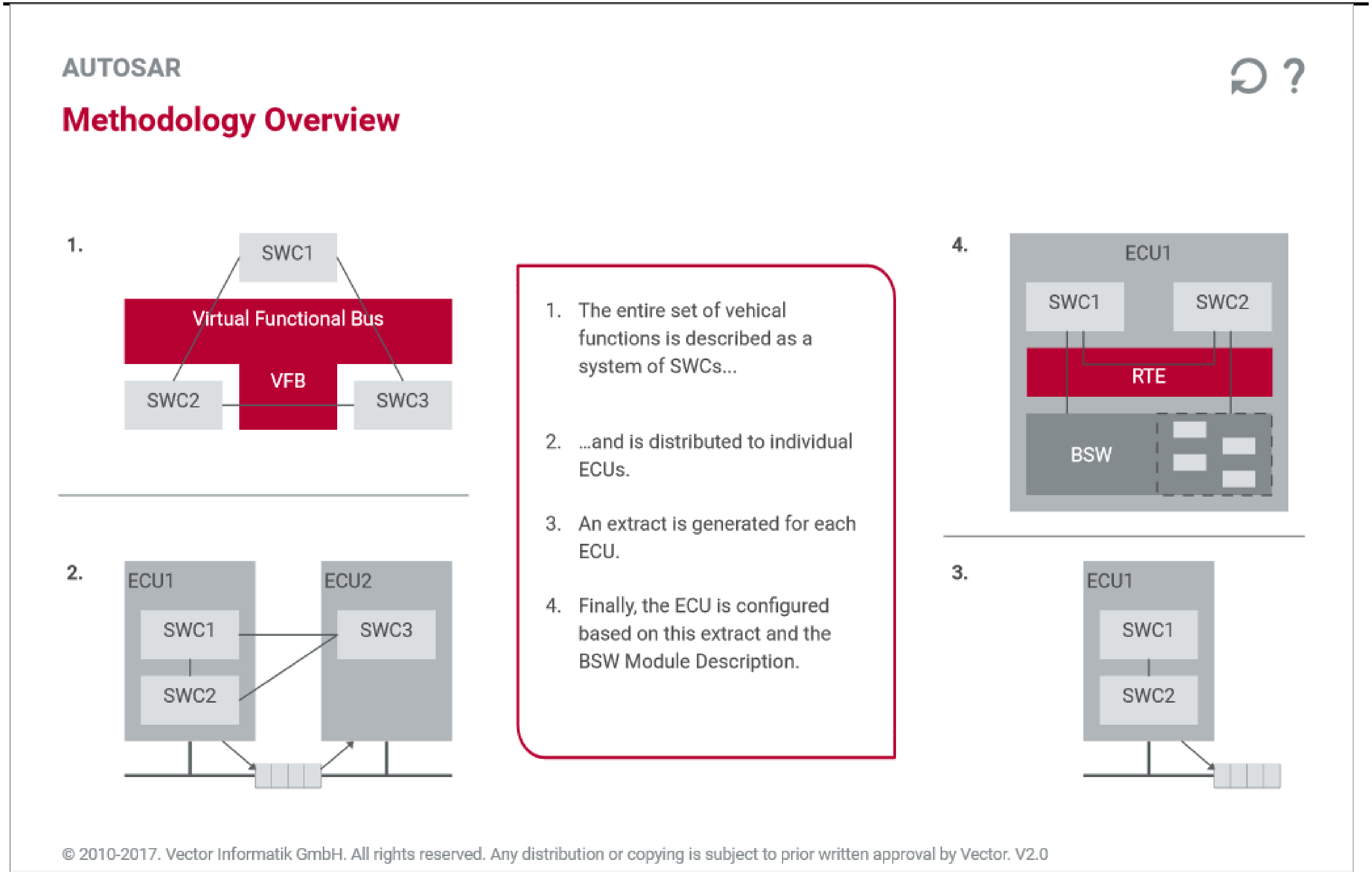
VC IVI Development Center Vietnam

Hanoi, Nov 2018

Be First, Do It Right, Work Smart

LG
Life's Good

Overview



AUTOSAR

Methodology Overview

1. The entire set of vehical functions is described as a system of SWCs...
2. ...and is distributed to individual ECUs.
3. An extract is generated for each ECU.
4. Finally, the ECU is configured based on this extract and the BSW Module Description.

Overview

Sys Config Input: Prepare the following:

- Software Components: each software component requires a description of the software API e.g. data types, ports, interfaces, etc…

- ECU Resources: each ECU requires specifications regarding e.g. the processor unit, memory, peripherals, sensors and actuators, …

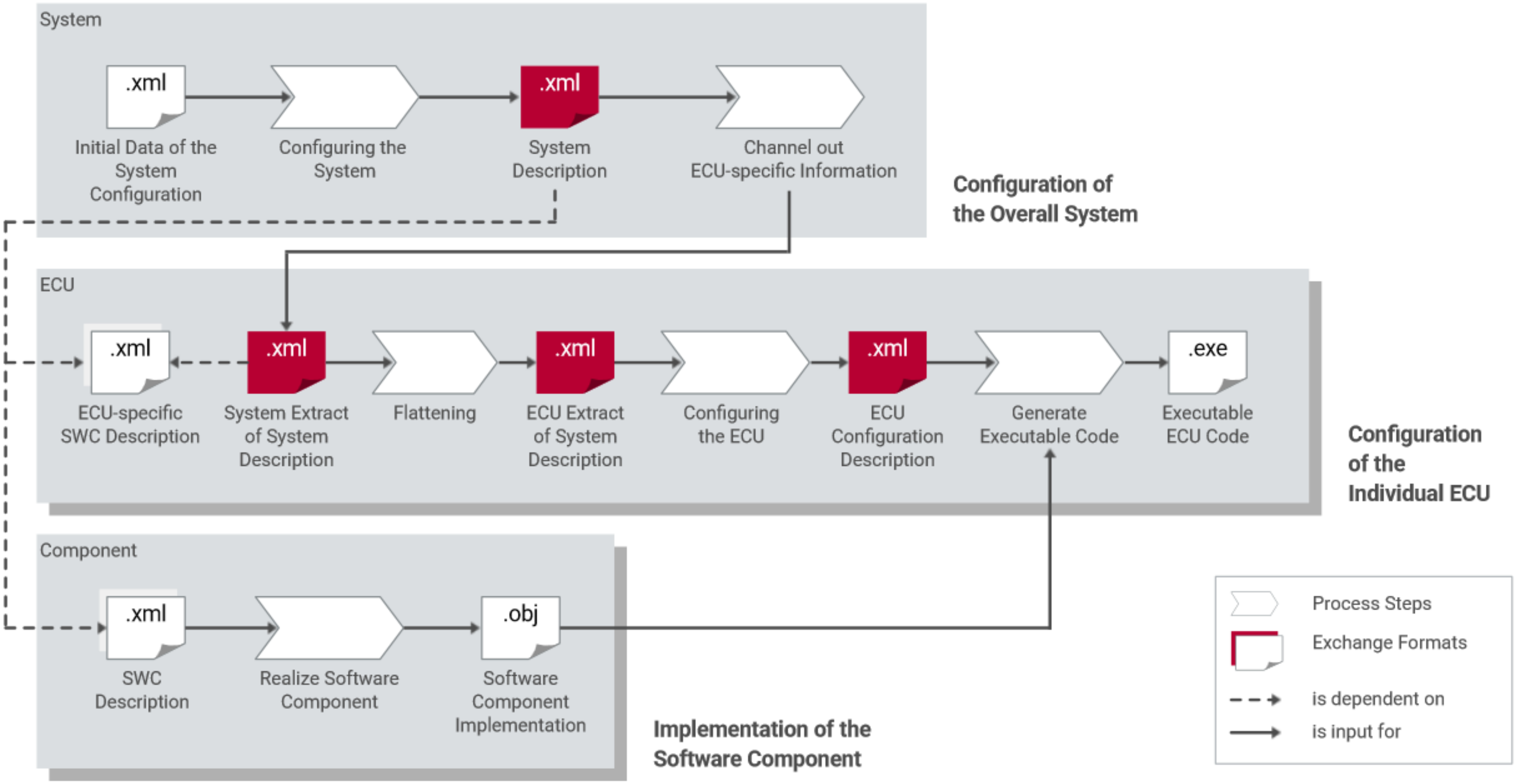
- System Constraints: this contains the bus signals, topology and mapping, …

- The activity of the Configure System mainly maps the software components to the ECUs with regard to resources and timing requirements.
- The output of the Configure System is the System Configuration Description. This description includes all system information (e.g. bus mapping, topology) and the mapping of which software component is located on which ECU.
- The step Extract ECU-Specific Information extracts the information from the System Configuration Description needed for a specific ECU. This is then placed in the ECU Extract of System Configuration.
- The activity Configure ECU adds all necessary information for implementation like task scheduling, necessary BSW (basic software) modules, configuration of the BSW, assignment of runnable entities to tasks, etc.
- The result of the activity Configure ECU is included in the ECU Configuration Description, which collects all information that is local to a specific ECU. The executable software to this specific ECU can be built from this information.

Overview

AUTOSAR

Methodology Levels | AUTOSAR 4



System

.xml Initial Data of the System Configuration → Configuring the System → .xml System Description → Channel out ECU-specific Information

Configuration of the Overall System

ECU

.xml ECU-specific SWC Description

.xml System Extract of System Description → Flattening → .xml ECU Extract of System Description → Configuring the ECU → .xml ECU Configuration Description → Generate Executable Code → .exe Executable ECU Code

Configuration of the Individual ECU

Component

.xml SWC Description → Realize Software Component → .obj Software Component Implementation

Implementation of the Software Component

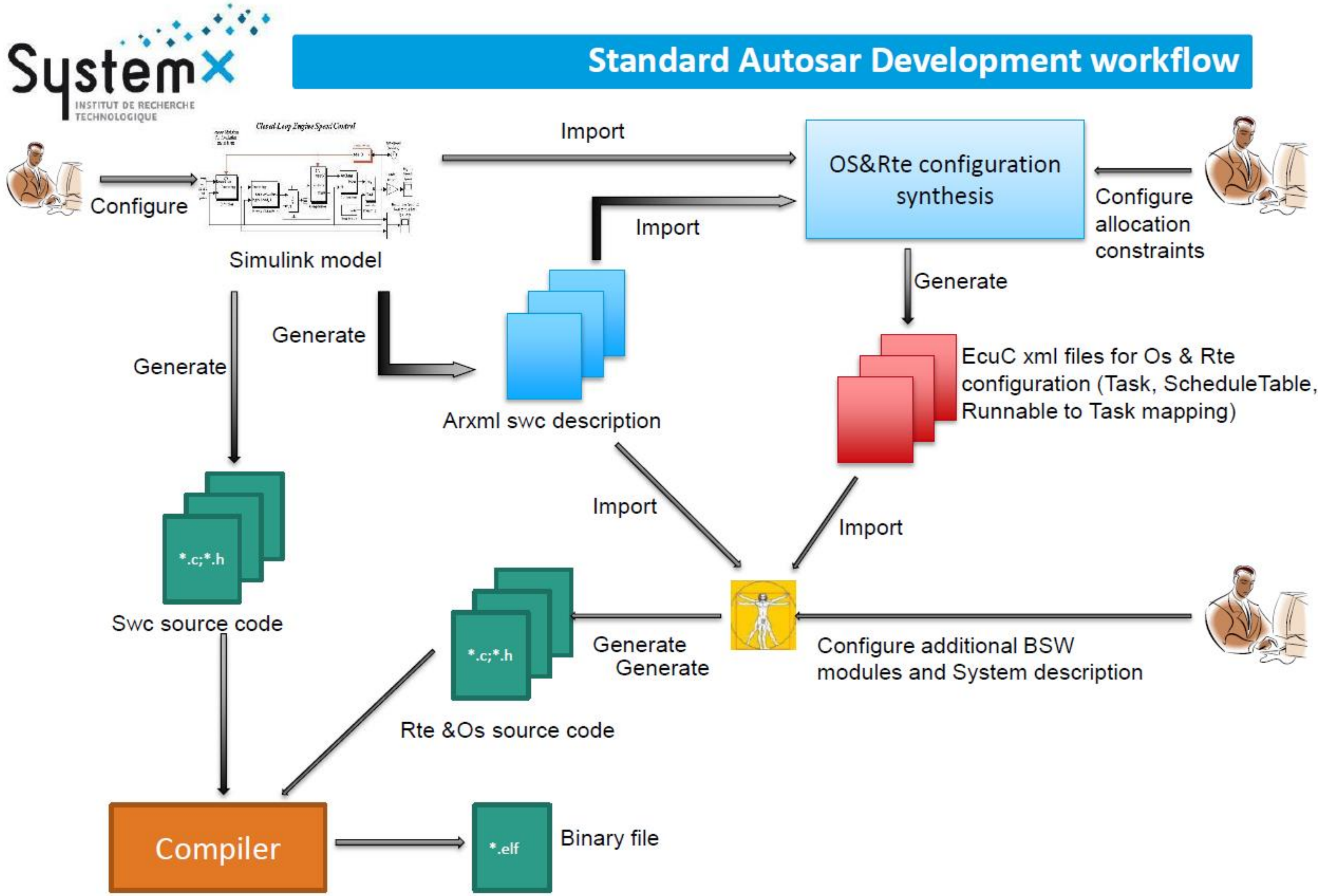
Process Steps

Exchange Formats

- - → is dependent on

→ is input for

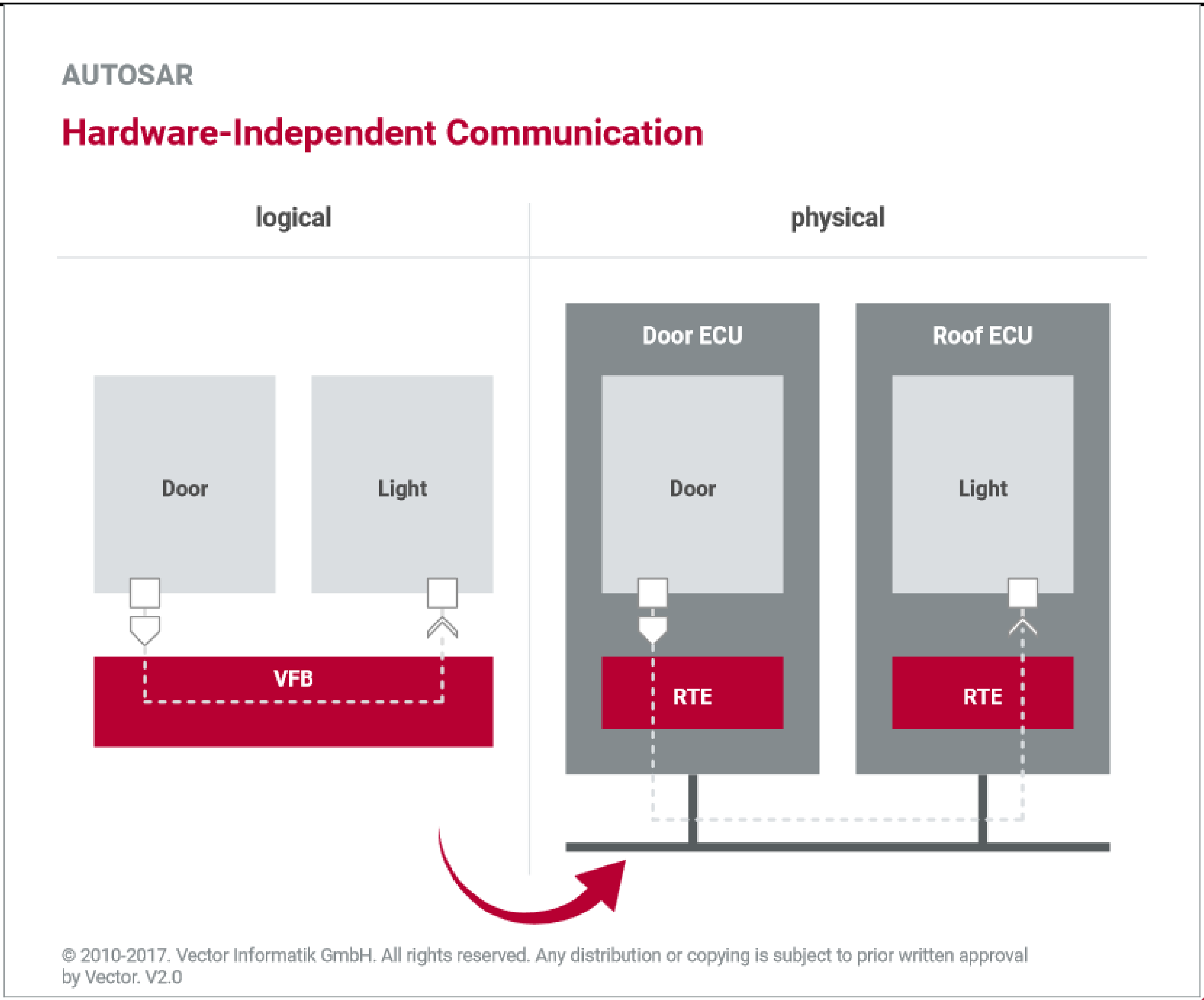
Overview


SystemX
INSTITUT DE RECHERCHE TECHNOLOGIQUE
Standard Autosar Development workflow
Configure
Closed-Loop Engine Speed Control
Simulink model
Import
OS&Rte configuration synthesis
Configure allocation constraints
Import
Generate
Generate
Generate
Arxml swc description
EcuC xml files for Os & Rte configuration (Task, ScheduleTable, Runnable to Task mapping)
Import
Import
.c;.h
Swc source code
.c;.h
Generate
Generate
Configure additional BSW modules and System description
Rte &Os source code
Compiler
*.elf
Binary file

AUTOSAR Exchange Formats

- SWC Description:
  The supplier or OEM defines the SWCs. This involves creating an XML file for each SWC: the SWC Description. It describes the interfaces and the resource requirements of the SWC. The supplier or OEM then creates the relevant C files for the implementation.
- System Description:
  The OEM first defines, based on the SWCs, the functional content for the entire vehicle independent of the ECUs. The OEM then designs the communication networks and distributes the SWCs to the existing ECUs. The results are saved in the system description.
- System Extract of System Description:
  For each ECU, the OEM reduces the System Description to a System Extract of System Description, which the OEM can share with the suppliers of the relevant ECU. This file replaces the .dbc, FIBEX or .ldf files that were previously used for configuration of the BSW modules.
- ECU Configuration Description (ECUC):
  The initial ECU Configuration Description is created by the supplier based on the current ECU Extract of System Description and the BSW Module Description files. The supplier then configures the ECU, using the ECU Configuration Description for documentation. This involves tools for setting and checking the parameters of the BSW modules and the RTE. The ECU Configuration Description is used as the foundation of the ECU-specific generation of RTE and BSW modules by the associated generators.
- Flow

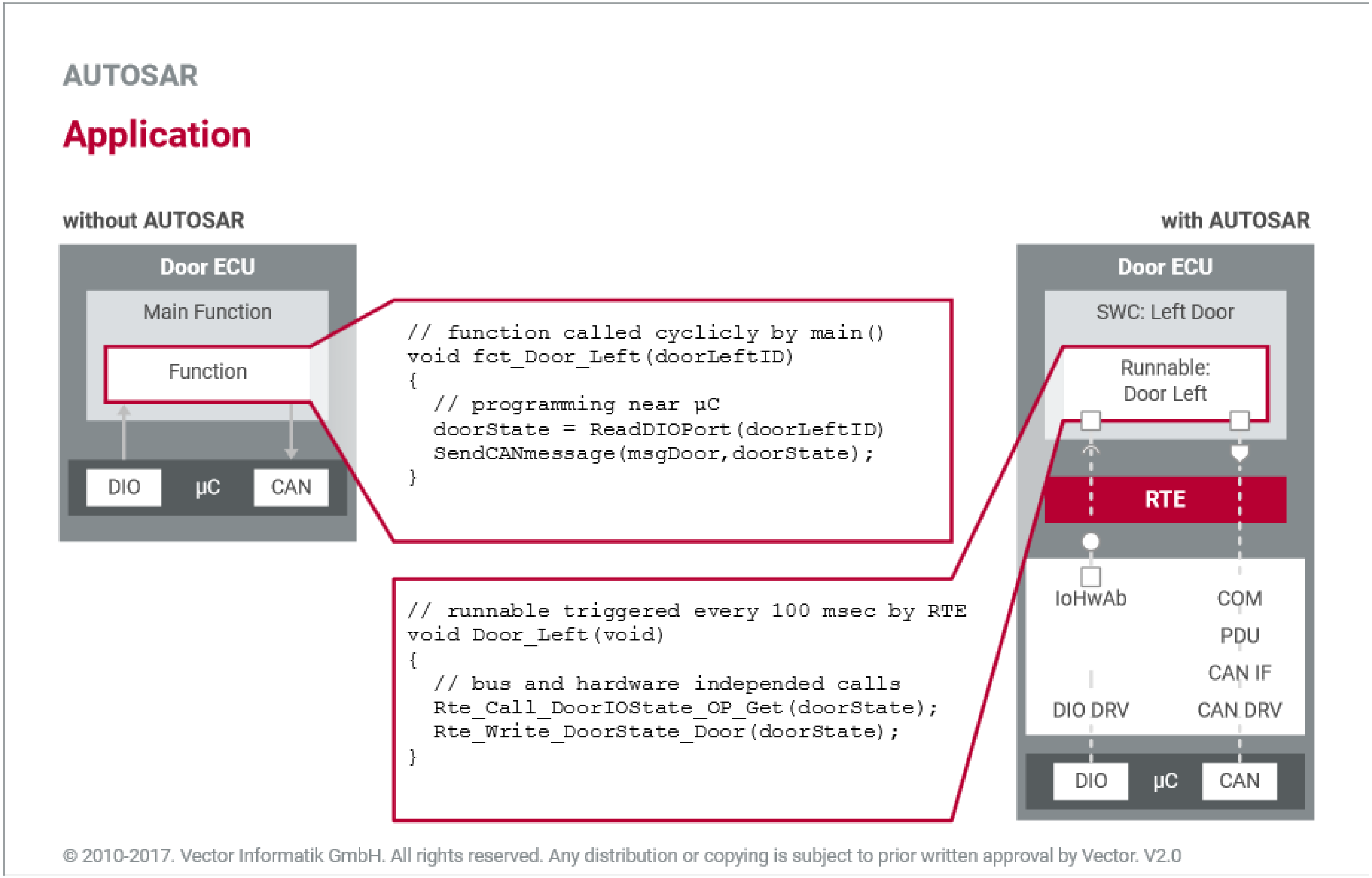
Development of the Functional Software



AUTOSAR

Hardware-Independent Communication

Development of the Functional Software



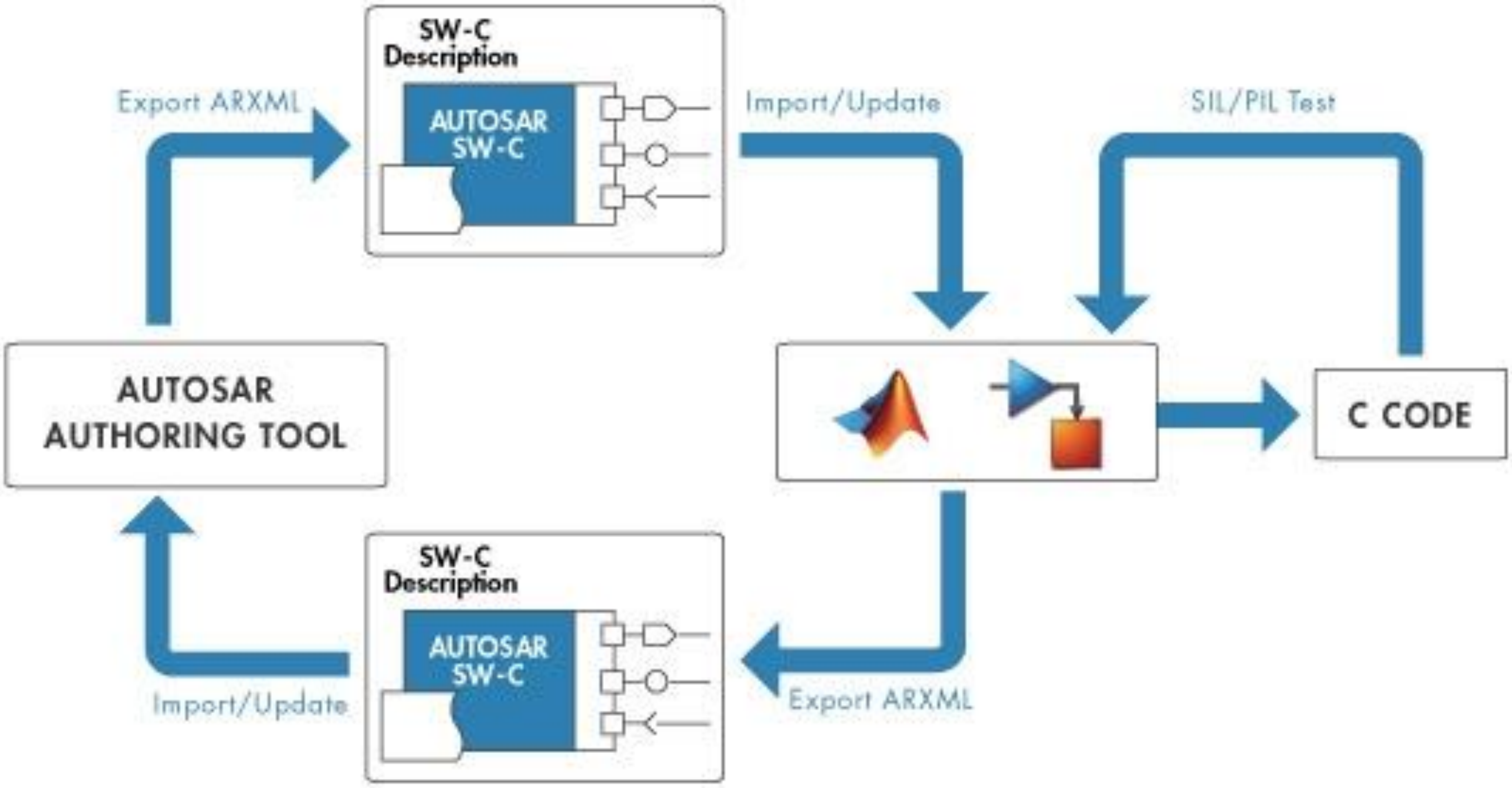
AUTOSAR

Application

```
// function called cyclicly by main()
void fct_Door_Left(doorLeftID)
{
  // programming near µC
  doorState = ReadDIOPort(doorLeftID)
  SendCANmessage(msgDoor,doorState);
}
```

```
// runnable triggered every 100 msec by RTE
void Door_Left(void)
{
  // bus and hardware independed calls
  Rte_Call_DoorIOState_OP_Get(doorState);
  Rte_Write_DoorState_Door(doorState);
}
```

Example: Using Matlab Simulink

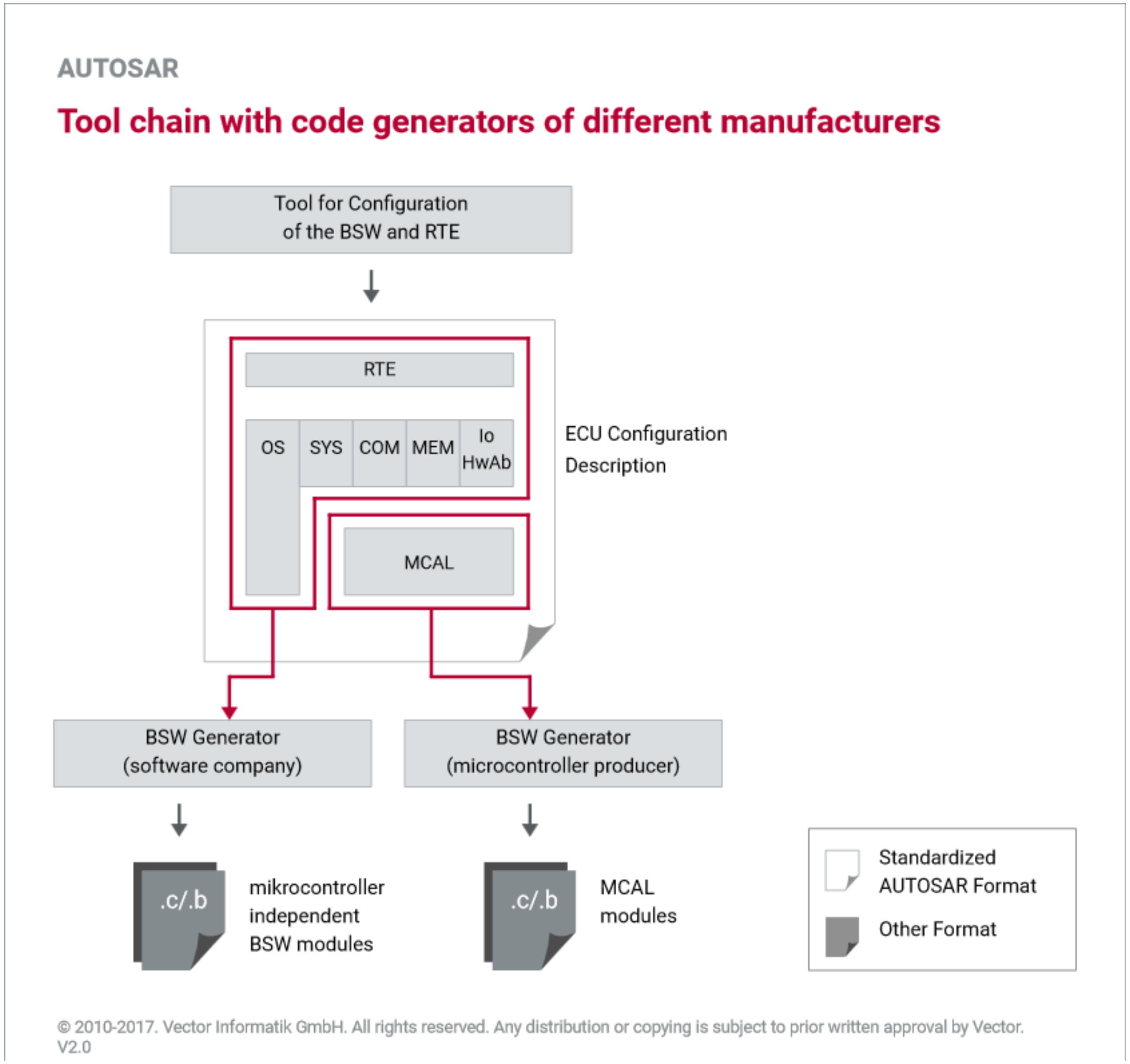


Example: Using Matlab Simulink

Key products for developing AUTOSAR applications:
Simulink and Stateflow® for software design
Embedded Coder for generating production code and ARXML, and for SIL/PIL verification
Polyspace Code Prover for verifying code of AUTOSAR Software Components

Third-party tools for AUTOSAR authoring: Vector Informatik DaVinci Developer, KPIT Cummins K-SAR, Mentor Graphics Volcano Vehicle System Architect, ETAS ISOLAR-A

Basic Software Producing

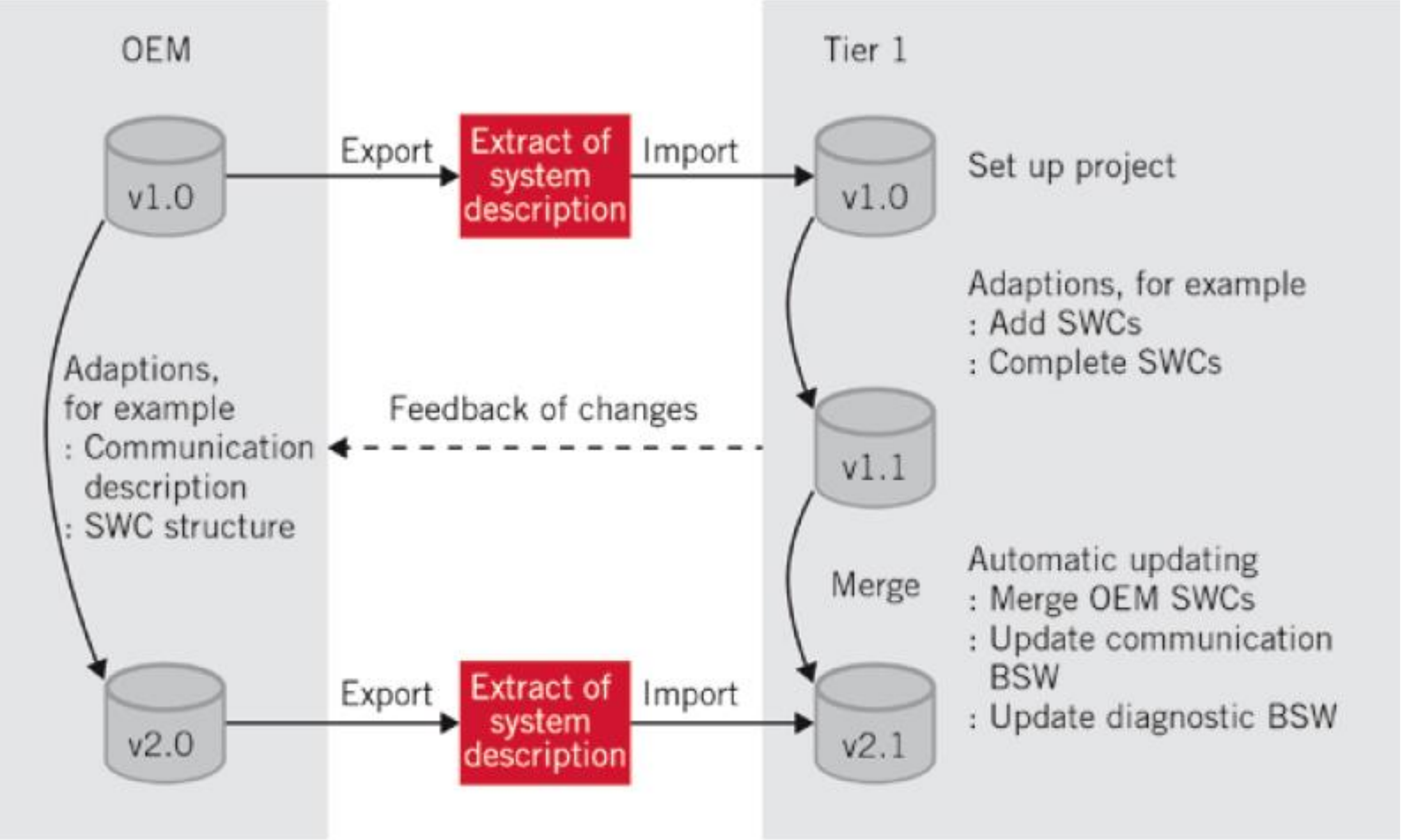


AUTOSAR

Tool chain with code generators of different manufacturers

Tool for Configuration of the BSW and RTE
RTE
OS
SYS
COM
MEM
Io HwAb
MCAL
ECU Configuration Description
BSW Generator (software company)
BSW Generator (microcontroller producer)
.c/.b
mikrocontroller independent BSW modules
.c/.b
MCAL modules
Standardized AUTOSAR Format
Other Format

Basic Software Producing



Thanks friends!